Power of Distributed Quantum Merlin-Arthur Proofs

Harumichi Nishimura (Nagoya U)

Based on arXiv: 2002.10018 (Proc. ITCS2021)

(joint work with P. Fraigniaud, F. Le Gall, Ami Paz)

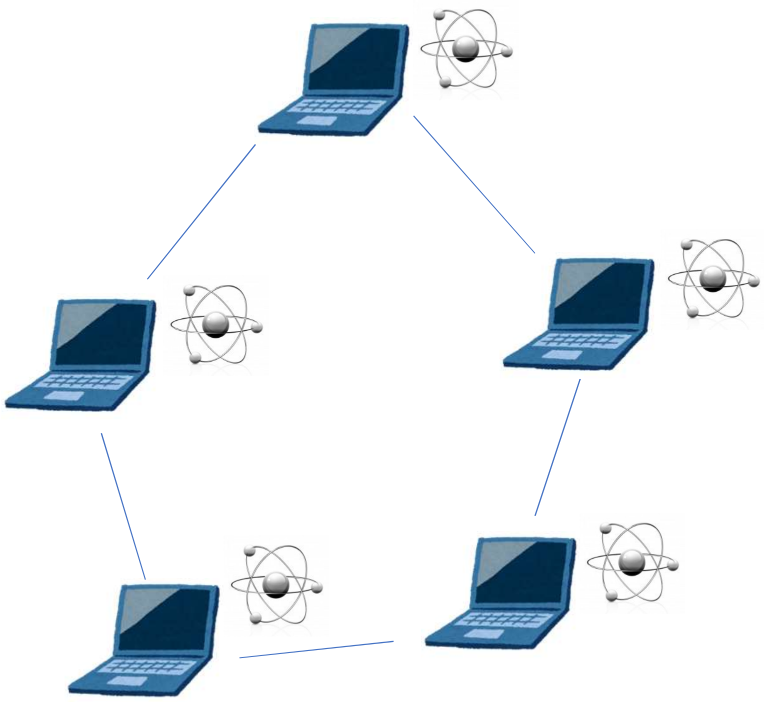
SUSTech-Nagoya workshop on Quantum Science 2022

June 2, 2022

Quantum Distributed Computing

- Leader election [Tani, Kobayashi, Matsumoto 05, 09]
- Byzantine agreement [Ben-Or, Hassidim 05]
- Diameter [Le Gall, Magniez 18]
- All pairs shortest paths [Izumi, Le Gall 19]
- Triangle finding [Izumi, Le Gall, Magniez 20]

etc

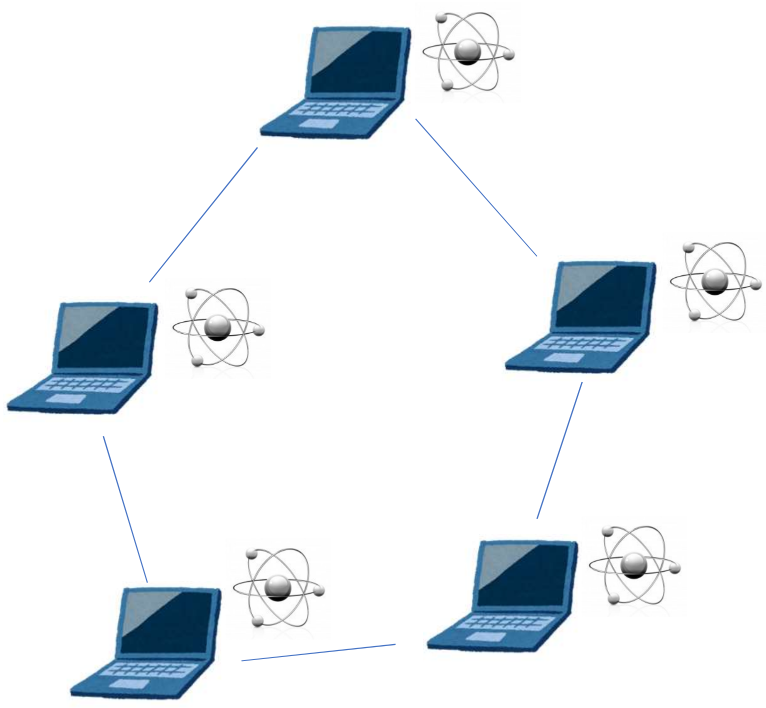


Quantum Distributed Computing

- Leader election [Tani, Kobayashi, Matsumoto 05, 09]
- Byzantine agreement [Ben-Or, Hassidim 05]
- Diameter [Le Gall, Magniez 18]
- All pairs shortest paths [Izumi, Le Gall 19]
- Triangle finding [Izumi, Le Gall, Magniez 20]

etc

- Our work: Distributed certification

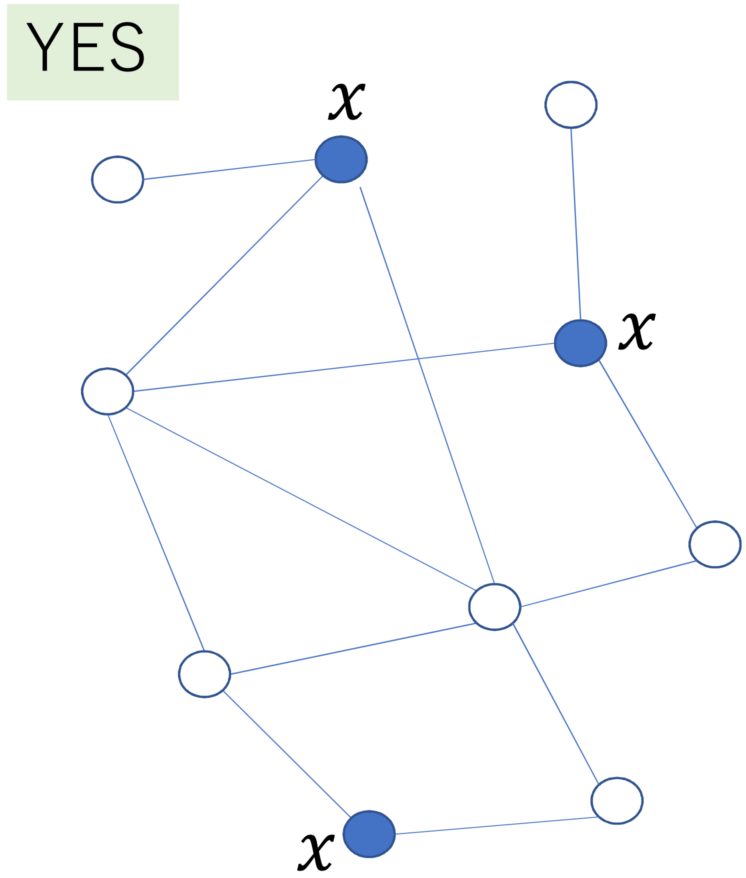


Outline

- Problem (Equality of data on networks)
- Setting (Distributed Merlin-Arthur protocols)
- Results (Quantum dMA protocols)
- Overview of our protocol

Our Problem: Equality of Data

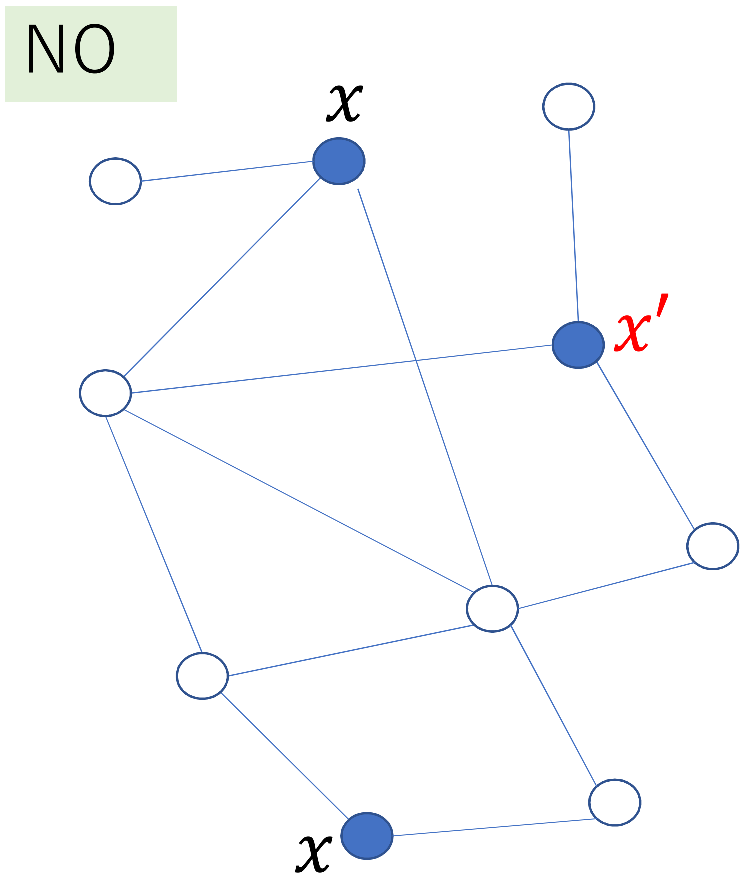
- Replicated data on a network
- Are all data identical?



terminals (nodes who have data)

Our Problem: Equality of Data

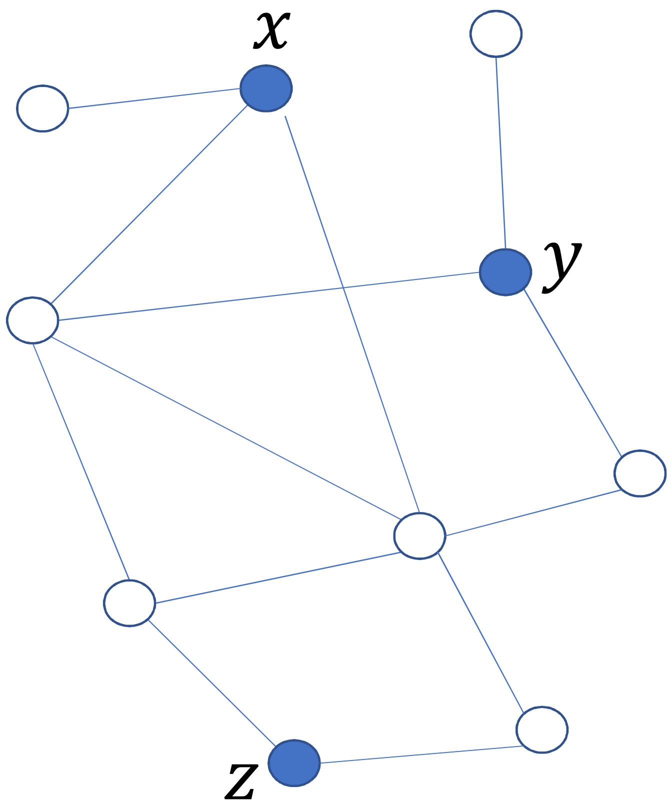
- Replicated data on a network
- Are all data identical?




terminals (nodes who have data)

Our Problem: Equality of Data

- Replicated data on a network
- Are all data identical?
- No O(1) round protocol
  - $\Omega(r)$ rounds are needed
    (r : diameter of the network)
  - We assume the nodes do not share prior randomness & entanglement
- ∃ 1 round “NP-like” protocol
  (distributed certification)




terminals (nodes who have data)

Outline

- Problem (Equality of data on networks)
- Setting (Distributed Merlin-Arthur protocols)
- Results (Quantum dMA protocols)
- Overview of our protocol

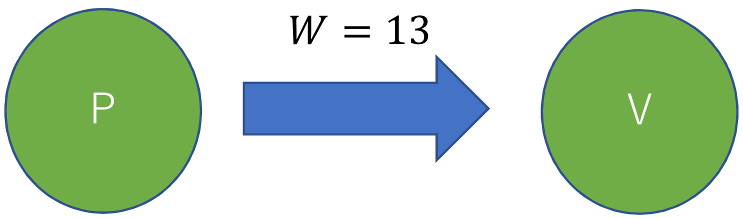
Merlin-Arthur Protocols

- Protocol between prover (Merlin) and verifier (Arthur)
  - Merlin: powerful (computationally unbounded) but untrusted
  - Arthur: wants to check some property but less powerful (polynomial-time)

Ex. "N is composite?" has a Merlin-Arthur protocol

(Completeness) If N is composite, the verifier can check it easily by receiving a non-trivial divisor as certificate

(Soundness) If not, the verifier rejects any message from the prover

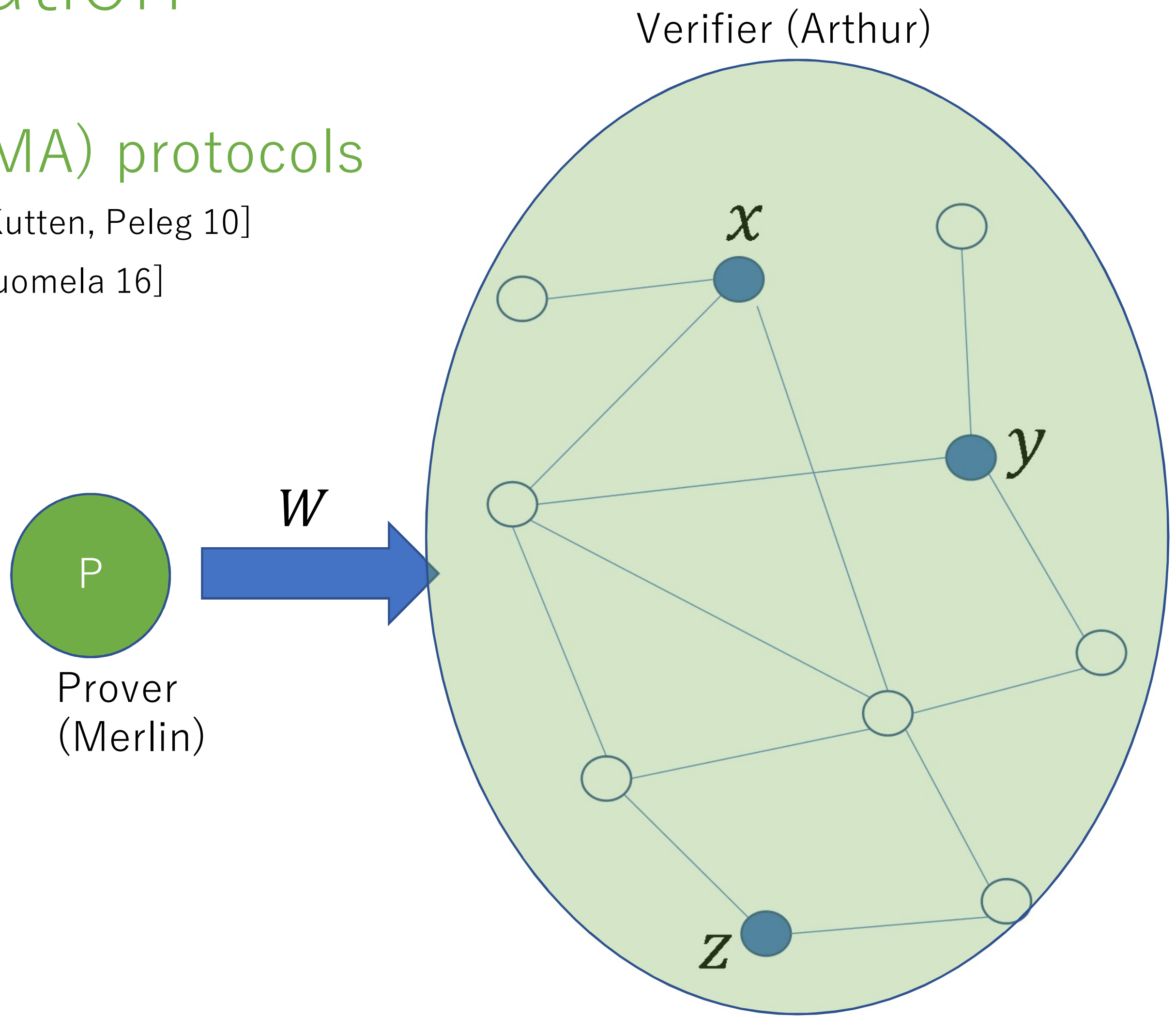


Input $N = 247$

Distributed Certification

- Distributed Merlin-Arthur (dMA) protocols
  - Proof labeling scheme [Korman, Kutten, Peleg 10]
  - Locally checkable proof [Goos, Suomela 16]

  etc



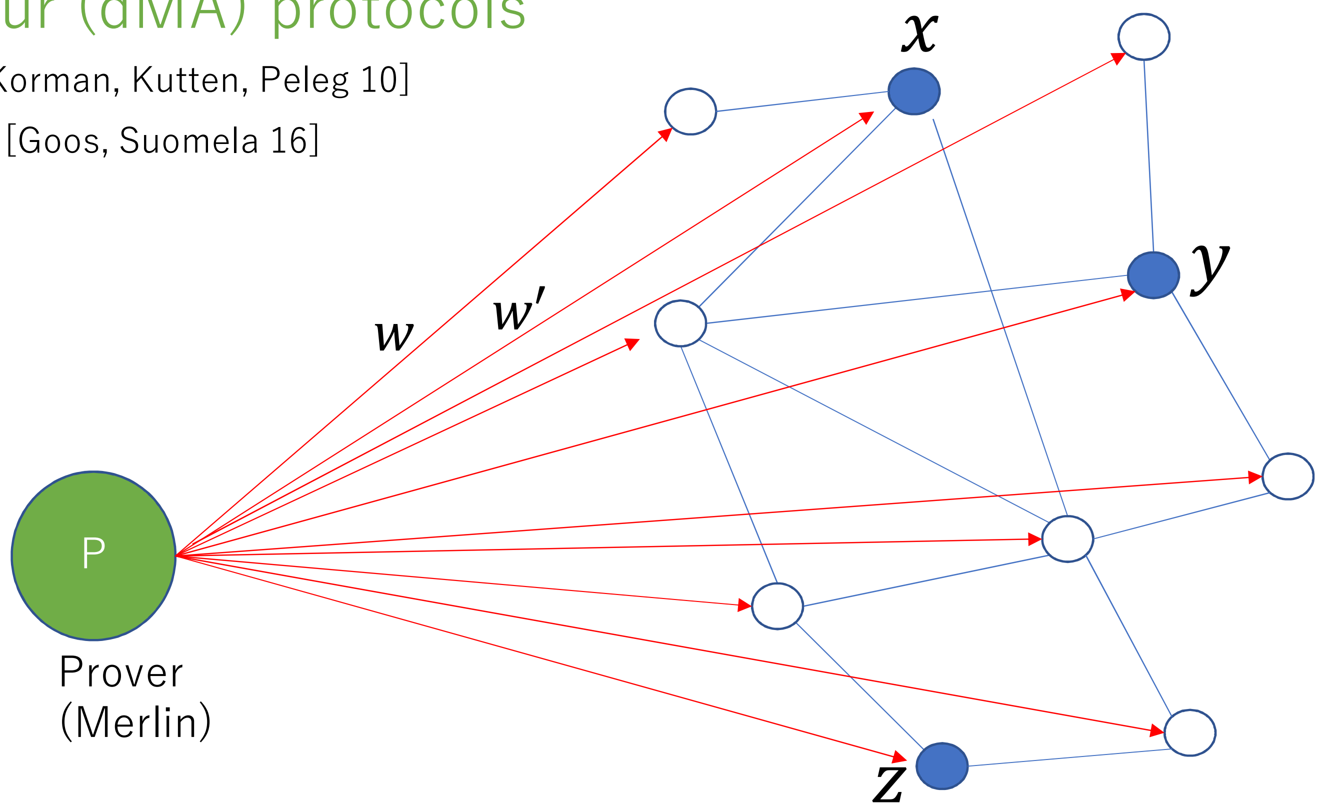
Distributed Certification

- Distributed Merlin-Arthur (dMA) protocols
  - Proof labeling scheme [Korman, Kutten, Peleg 10]
  - Locally checkable proof [Goos, Suomela 16]

  etc

Two phases:

1. (Prover phase) Prover sends certificates to each node



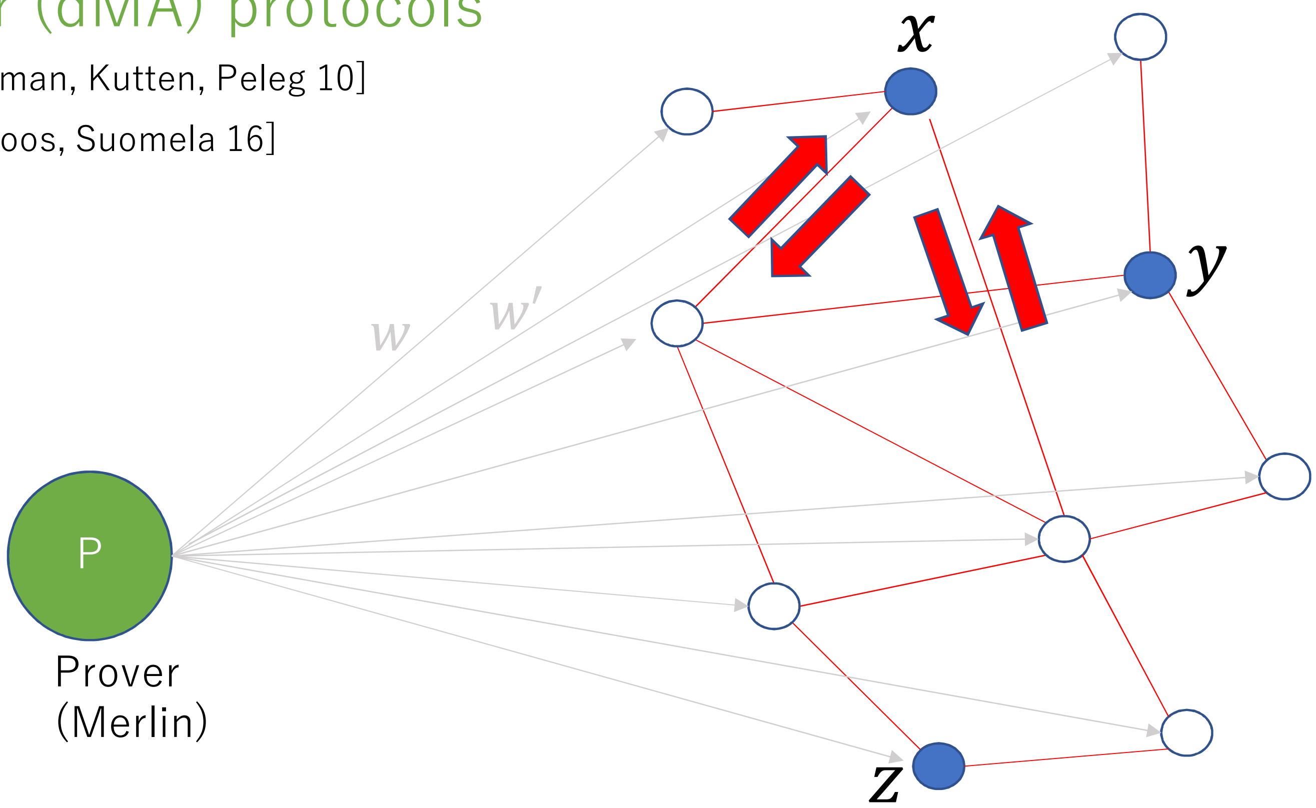
Distributed Certification

- Distributed Merlin-Arthur (dMA) protocols
  - Proof labeling scheme [Korman, Kutten, Peleg 10]
  - Locally checkable proof [Goos, Suomela 16]

  etc

Two phases:

1. (Prover phase) Prover sends certificates to each node
2. (Verification phase) Each node exchanges messages with the neighbors

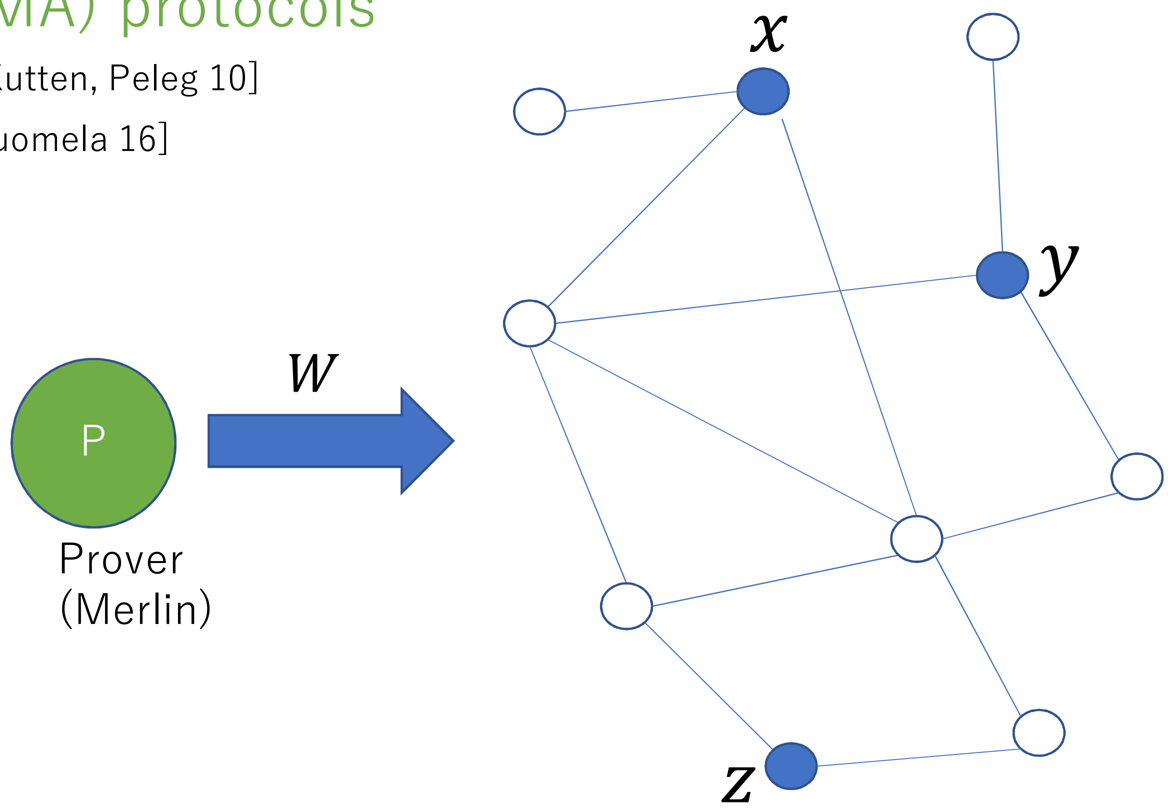


Distributed Certification

- Distributed Merlin-Arthur (dMA) protocols
  - Proof labeling scheme [Korman, Kutten, Peleg 10]
  - Locally checkable proof [Goos, Suomela 16]

  etc

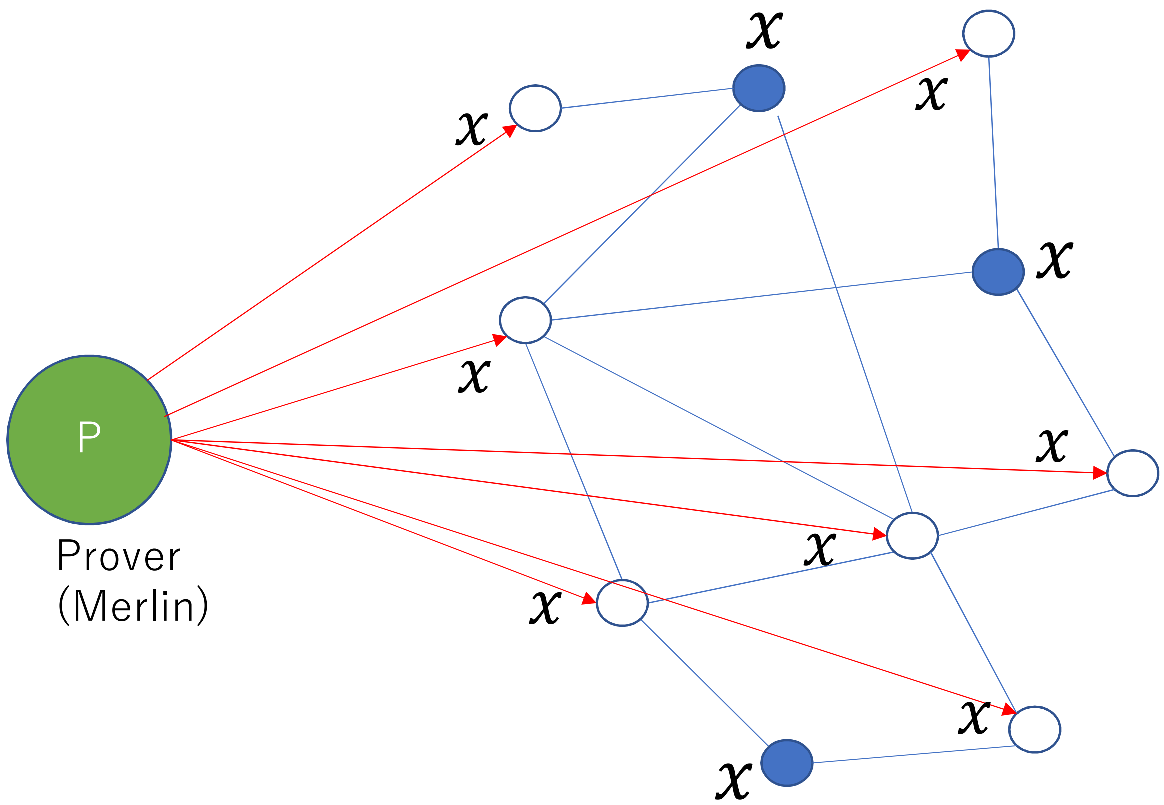
Properties:
(YES case: Completeness)
$\exists w$[all nodes accept]
(w.h.p.)
(NO case: Soundness)
$\forall w$[some node rejects]
(w.h.p.)



dMA Protocol for EQ of Data

Trivial protocol:
(P) Prover sends x when all data are x
(V) Each node checks if it is same as the neighbor's one

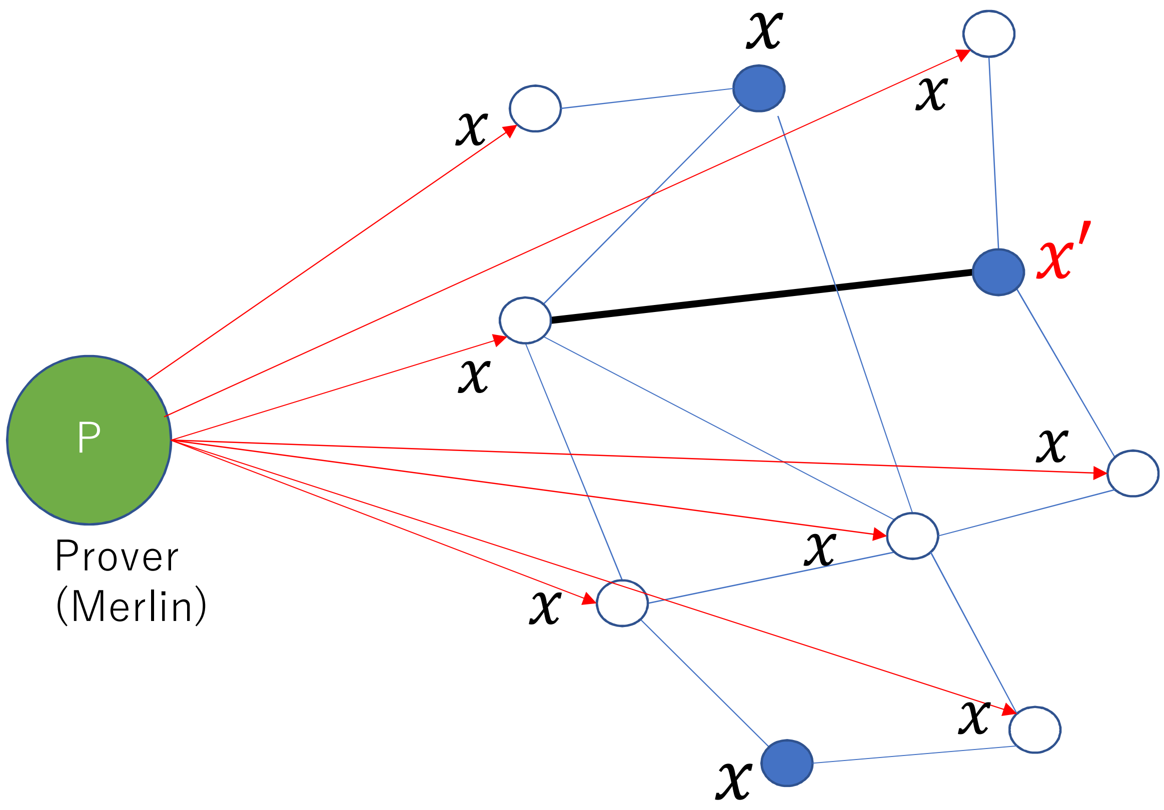
(YES case: Completeness)
$\exists W$ [all nodes accept]



dMA Protocol for EQ of Data

Trivial protocol:
(P) Prover sends x when all data are x
(V) Each node checks if it is same as the neighbor's one

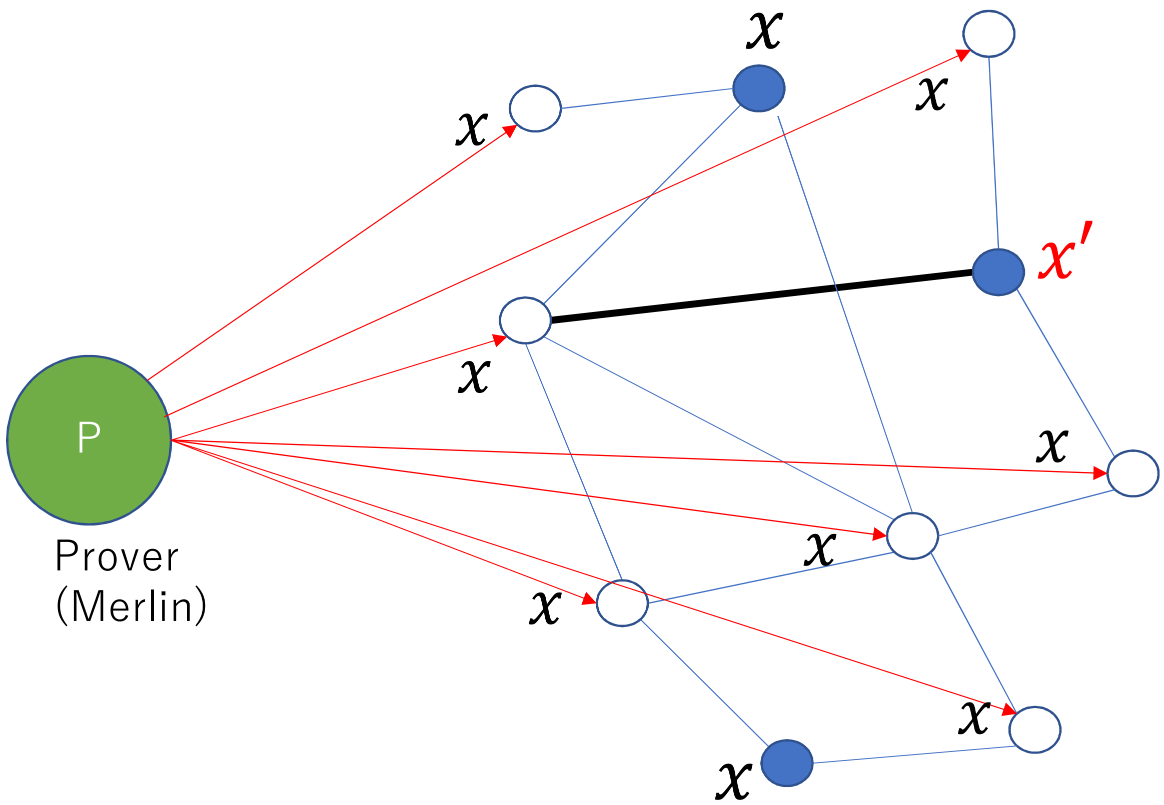
(NO case: Soundness)
$\forall W$[some node rejects]



dMA Protocol for EQ of Data

<u>Trivial Protocol is communication inefficient</u>

- Prover sends n bits for each node ($n := $ length of x)
- Each node sends n bits to the neighbors

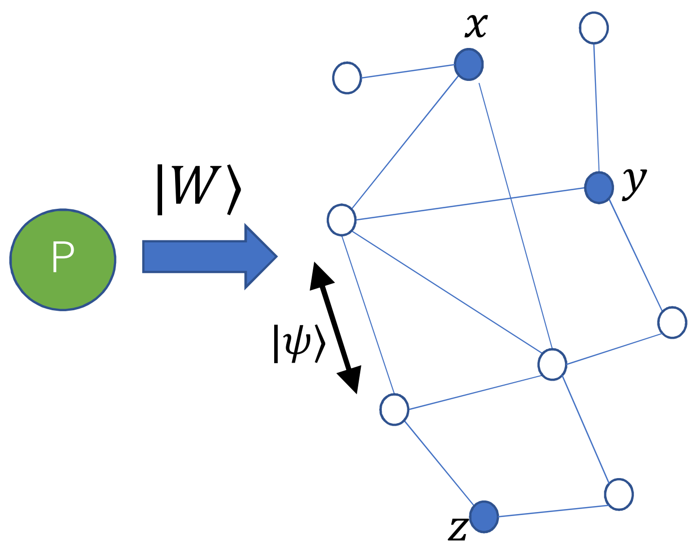


Outline

- Problem (Equality of data on networks)
- Setting (Distributed quantum Merlin-Arthur protocols)
- Results (Quantum dMA protocols)
- Overview of our protocol

Our Results

- Distributed Quantum Merlin-Arthur (dQMA) protocols for "Equality of Data" on the network
  - Quantum certificates from the prover
  - Quantum messages among nodes
- Classical lower bound
  - Any dMA protocol requires $\Omega(n)$-bit certificates if error probability is reasonably small (say, 1/3)
- Quantum upper bound
  - ∃ dQMA protocol for equality of replicated data with $O(tr^2 \log(n+r))$-qubit certificates & messages
    - t:= number of the terminals (= nodes who have data)
    - r ≔ diameter of the network
    - t and r are typically much smaller than n

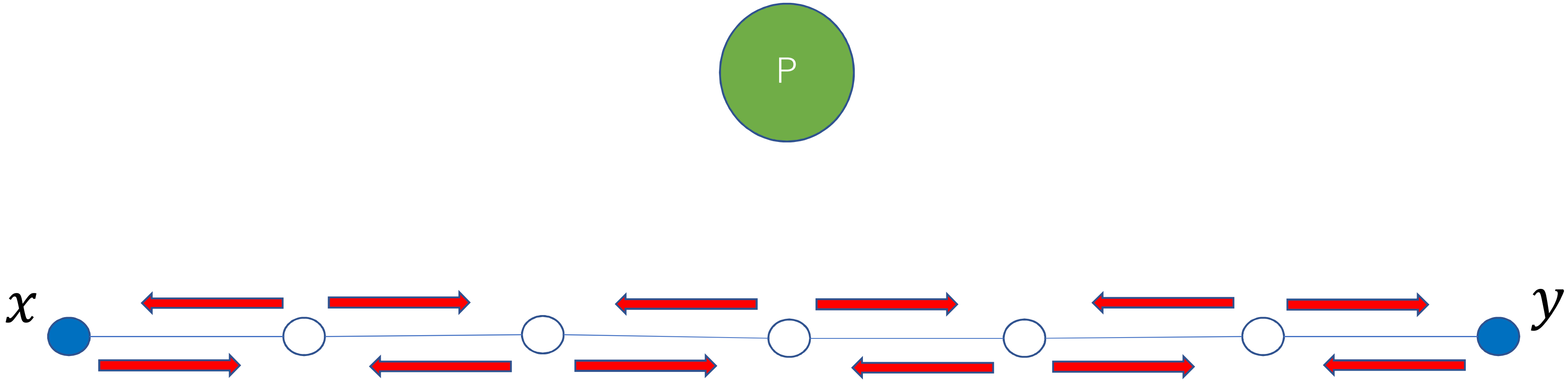


Outline

- Problem (Equality of data on networks)
- Setting (Distributed quantum Merlin-Arthur protocols)
- Results (Quantum dMA protocols)
- Overview of our protocol
  - Path networks

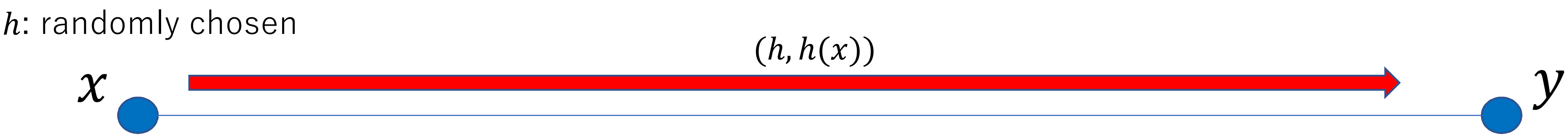
Path

- Path network
  - $t = 2$, r =path length
  - Only the left & right nodes have input strings



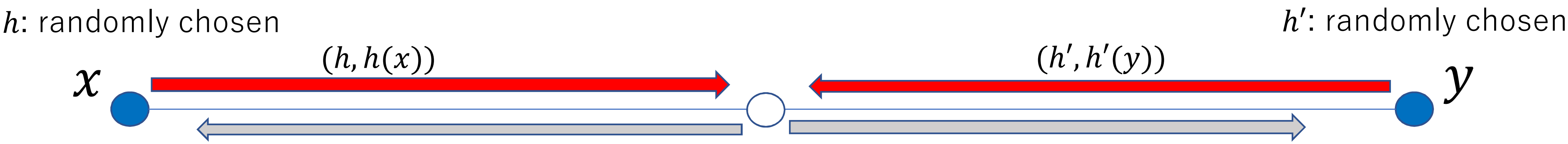
Path (2 nodes): Classical case

- $O(\log n)$ messages are enough on the path of 2 nodes
  - Prover is unnecessary
  - Use hash functions
    - $\Pr_h[h(x) \neq h(y)] \leq 1/\mathrm{poly}(n)$ when $x \neq y$
    - Length of pair $\left(h, h(x)\right) = O(\log n)$



Path (3 nodes or more): Classical case

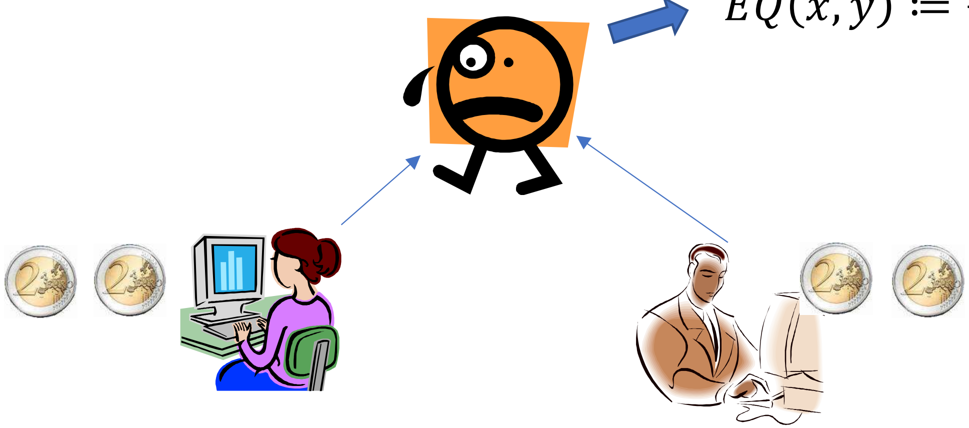
- Similar strategy is impossible on the path of 3 nodes as the left node and the right node cannot communicate directly in one round
- The case of 3 nodes is similar to the SMP model in communication complexity (since the central node has no information on inputs and his/her simultaneous message is useless)



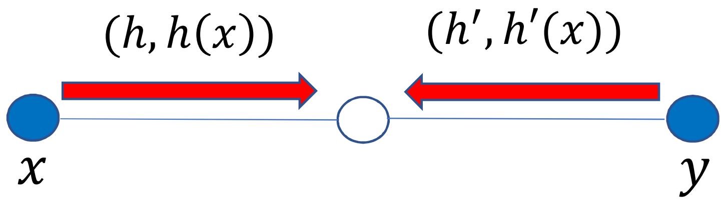
SMP complexity of EQ

- $CC^{smp}(EQ_n) = 2n$
- $RCC^{smp}(EQ_n) = \Theta(\sqrt{n})$ [Amb96,NS96,BK97]

$$EQ(x,y) := \begin{cases} 1 & (x = y) \\ 0 & (x \neq y) \end{cases}$$

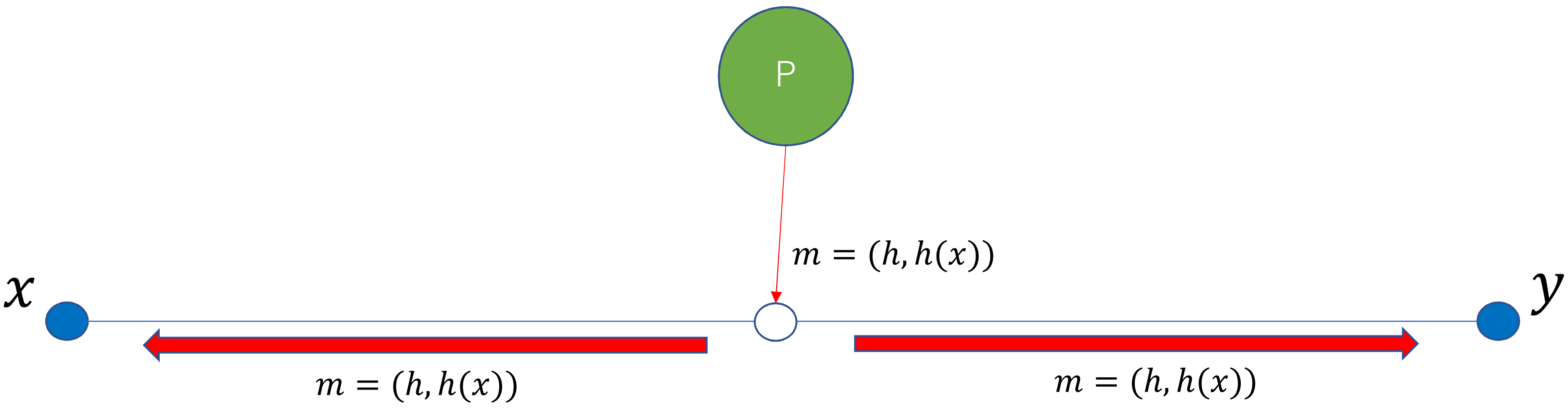


$x \in \{0,1\}^n$ $y \in \{0,1\}^n$



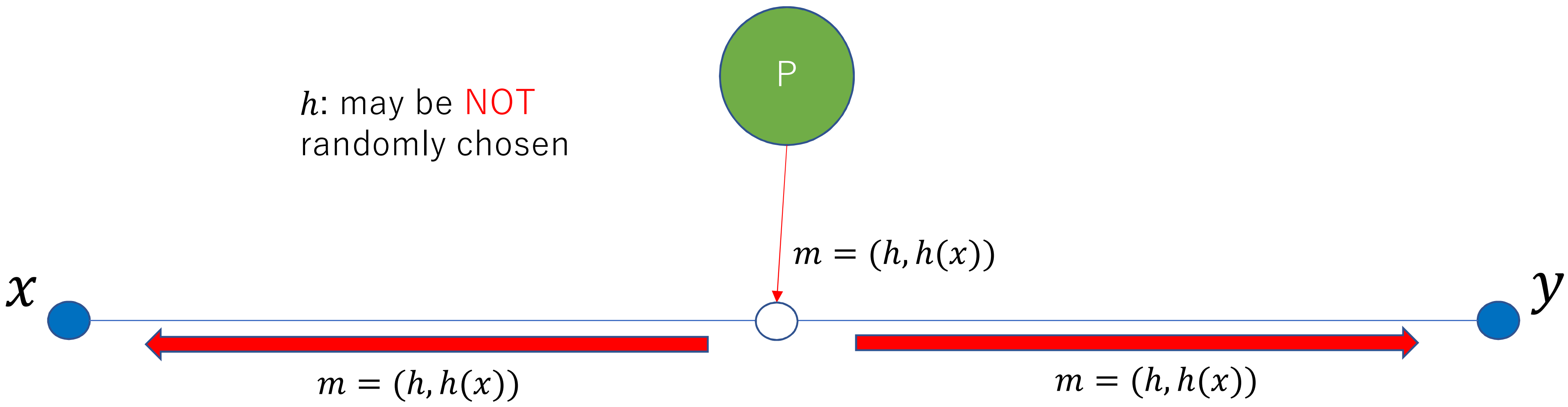
Path (3 nodes or more) with a prover

- How about the dMA-case (i.e., with the help of a prover)?



Path (3 nodes or more) with a prover

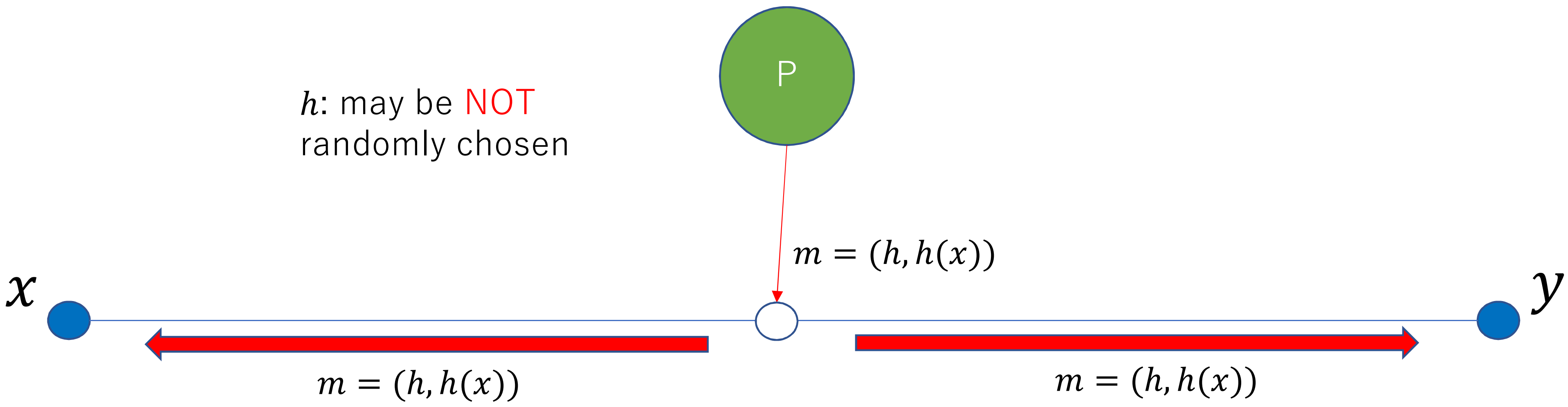
- How about the dMA-case (i.e., with the help of a prover)?
- Prover may be malicious



Path (3 nodes or more) with a prover

[Our classical lower bound]

Classical lower bound $\Omega(n)$ for the prover's certificate size can be proved for the path of 4 nodes

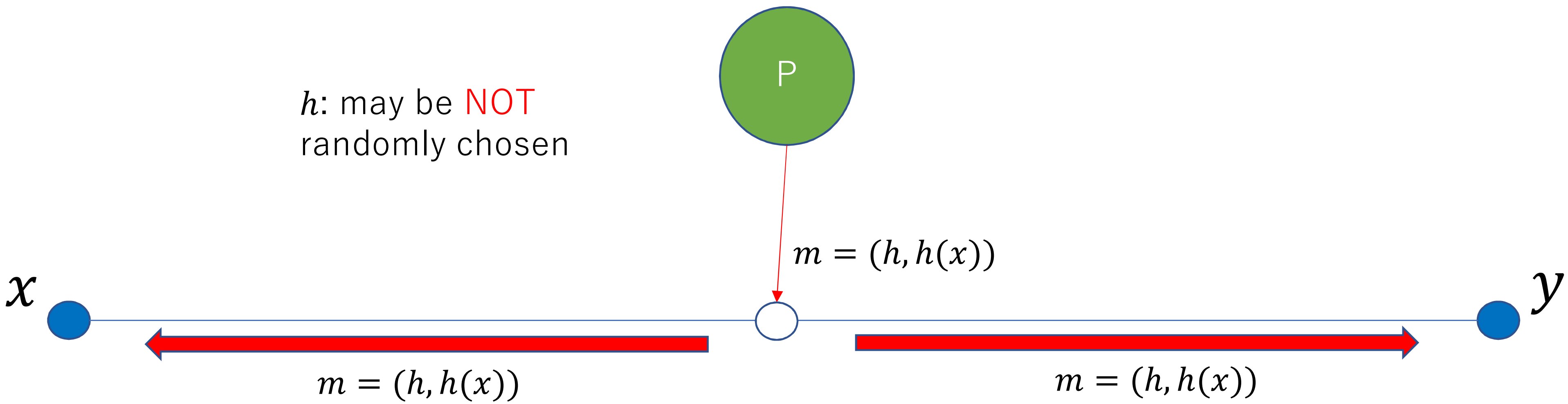


Path (3 nodes or more) with a prover

[Our classical lower bound]

Classical lower bound $\Omega(n)$ for the prover's certificate size can be proved for the path of 4 nodes

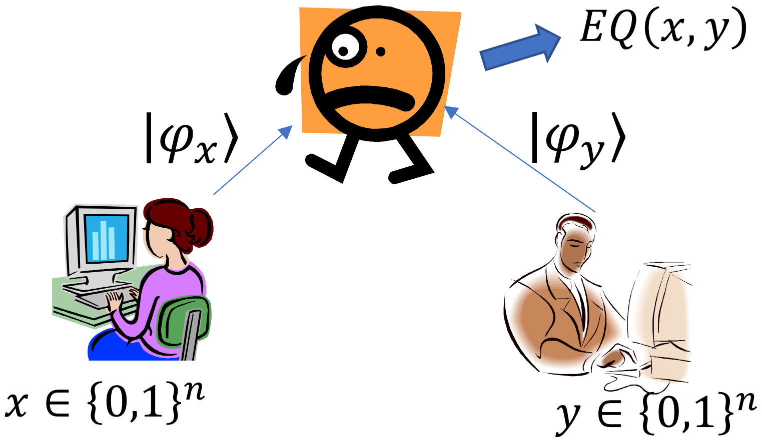
Q. How about the quantum MA protocols?



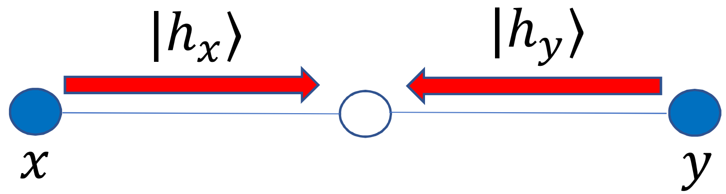
SMP complexity of EQ

- Classical Case
  - $CC^{smp}(EQ_n) = 2n$
  - $RCC^{smp}(EQ_n) = \Theta(\sqrt{n})$ [Amb96,NS96,BK97]
- Quantum Case
  - $QCC^{smp}(EQ_n) = O(\log n)$ [BCWW01]

Improvement by
quantum communication

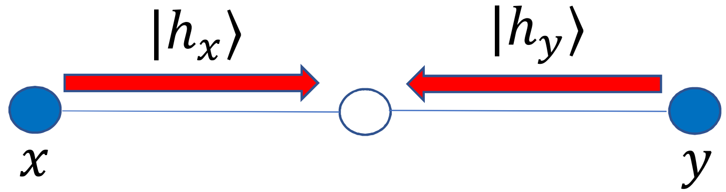


$$EQ(x, y) := \begin{cases} 1 & (x = y) \\ 0 & (x \neq y) \end{cases}$$

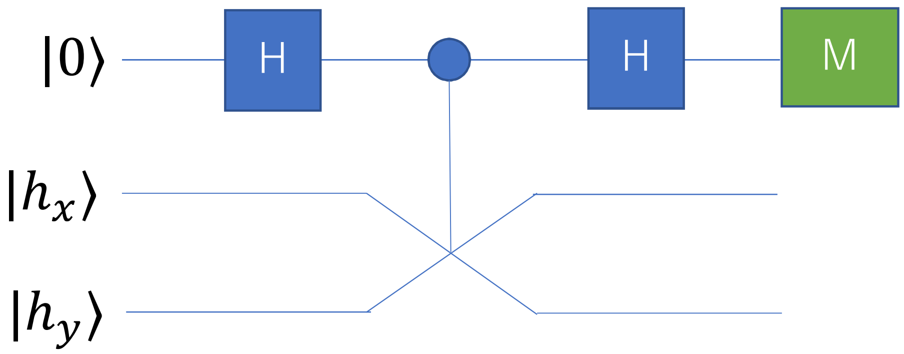


Basic Tools for Quantum Protocol

- Quantum fingerprint [Buhrman, Cleve, Watrous, de Wolf 01]
  - $|h_x\rangle = \sum_h |h\rangle|h(x)\rangle$ ($O(\log n)$-qubit state)
  - $|\langle h_x|h_y\rangle|^2 < 1/\mathrm{poly}(n)$ when $x \neq y$



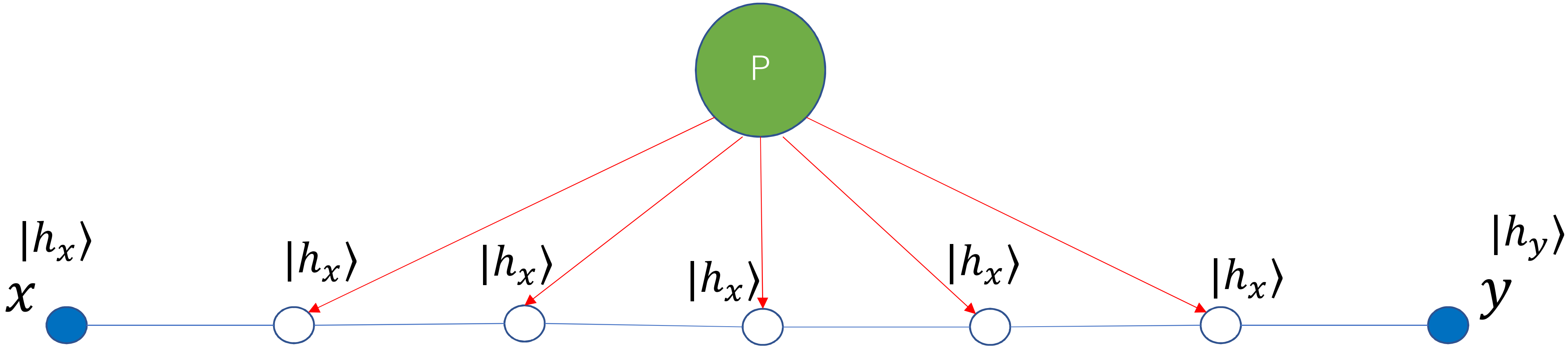
- SWAP test [Buhrman, Cleve, Watrous, de Wolf 01]
  - Can estimate $|\langle h_x|h_y\rangle|^2$ even if the input states $|h_x\rangle, |h_y\rangle$ are not known
  - $O(\log n)$ is enough for the 3 nodes case without the prover
- Our protocol uses quantum fingerprints as "certificates"



$$\Pr[M = 0\ (\text{accept})] = \frac{1}{2} + \frac{1}{2}|\langle h_x|h_y\rangle|^2$$

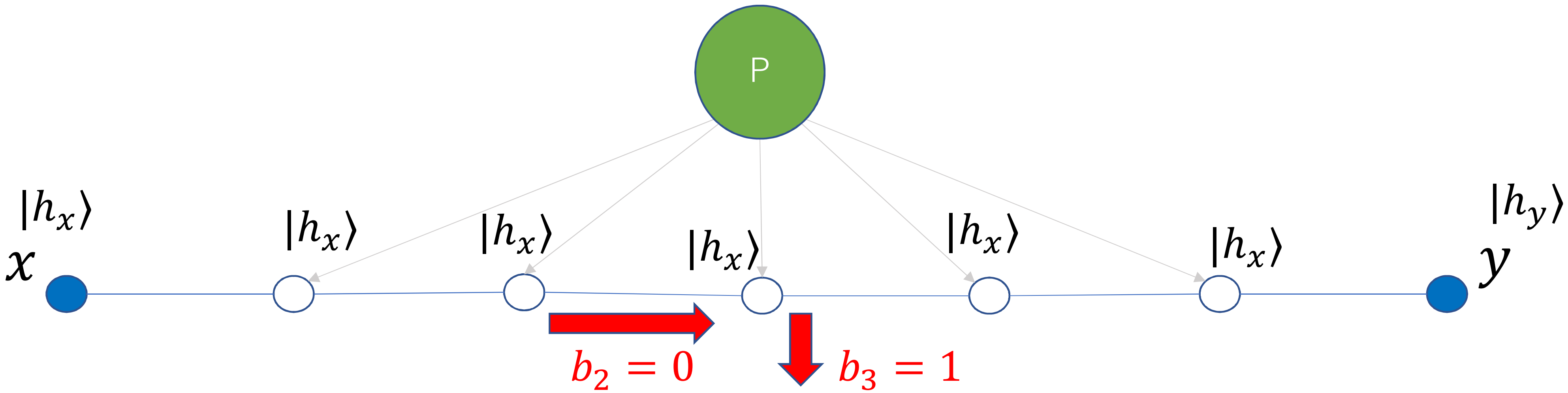
Our Quantum Protocol (Prover phase)

- Honest prover (when $x = y$) sends certificate $|h_x\rangle$ to each of the intermediate nodes
- The left node creates $|h_x\rangle$ and the right node creates $|h_y\rangle$



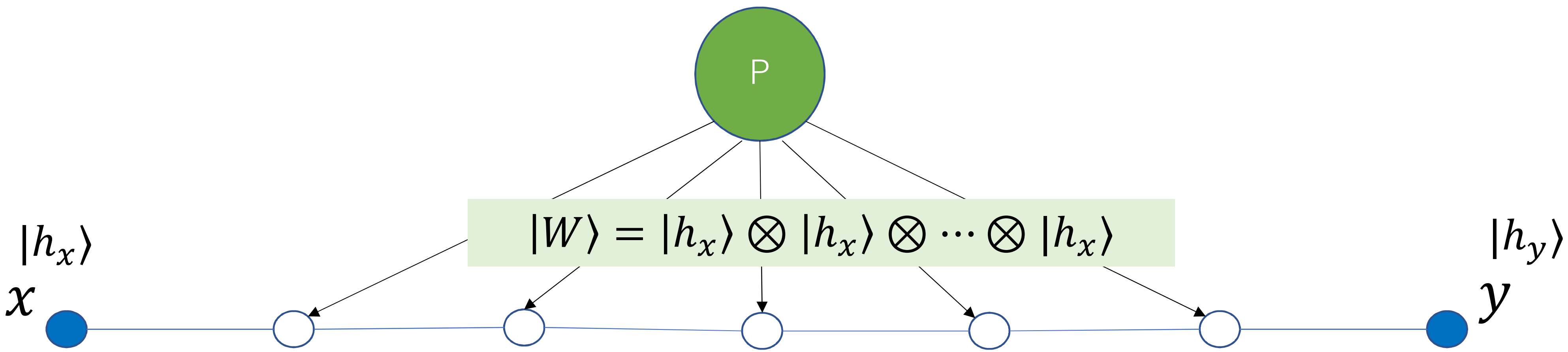
Our Quantum Protocol (Verification phase)

1. Each node j (except right node) chooses $b_j \in \{0,1\}$ uniformly at random: if $b_j = 0$, j sends the state to the right neighbor; otherwise, keep it by itself.
2. Each node (except left node) does SWAP test if it has two states, and outputs its result (accept/reject), and accepts otherwise



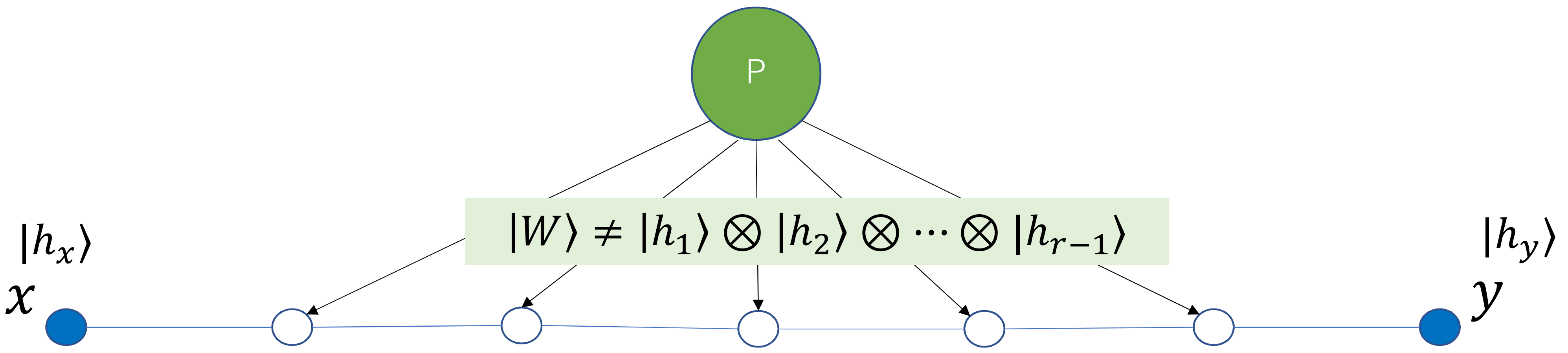
Analysis

- When $x = y$, all nodes accept with probability 1



Analysis

- When $x = y$, all nodes accept with probability 1
- When $x \neq y$, the probability that all nodes accept is $1 - \Omega(1/r^2)$
- Soundness error can be reduced to 1/3 by $O(r^2)$ repetitions

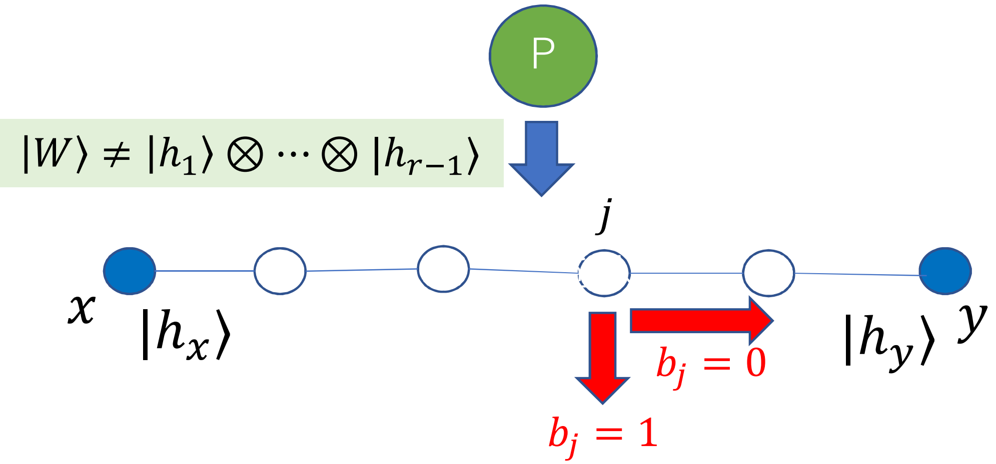


Soundness: $x \neq y$ (NO instance)

- We want some node to reject SWAP test with prob. $\Omega(1/r^2)$

Verification phase

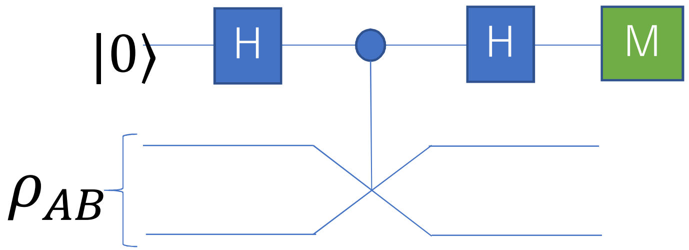
1. Each node j (except right node) chooses $b_j \in \{0,1\}$ uniformly at random: if $b_j = 0$, j sends the state to the right neighbor; otherwise, keep it by itself.
2. Each node (except left node) does SWAP test if it has two states, and outputs its result (accept/reject), and accepts otherwise



Soundness: $x \neq y$ (NO instance)

- We want some node to reject SWAP test with prob. $\Omega(1/r^2)$
- The property we use:

[Property] If the SWAP test accepts on input ρ_{AB} w.h.p., the two reduced states ρ_A & ρ_B must be close ($\rho_A \approx \rho_B$)



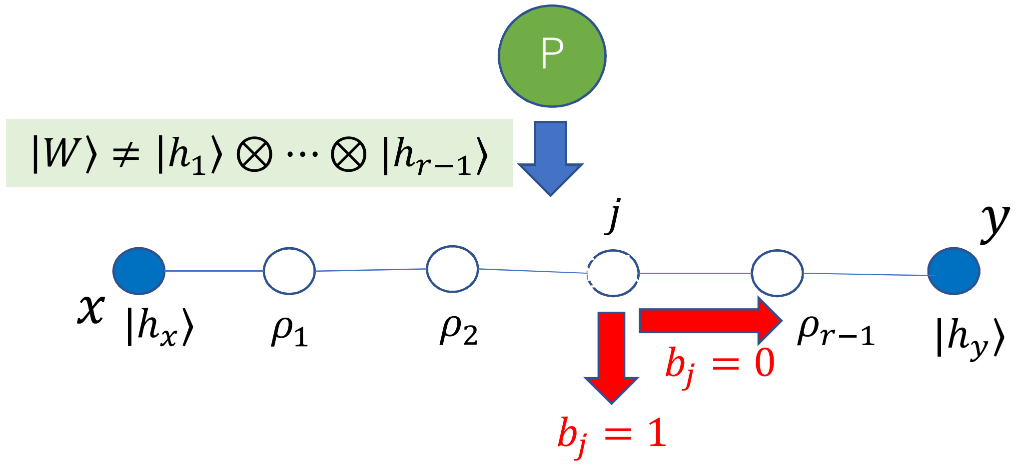
- Assuming all nodes accept w.h.p.,

$$|h_x\rangle \approx \rho_1 \approx \rho_2 \approx \cdots \approx \rho_{r-1} \approx |h_y\rangle,$$

which contradicts $|\langle h_x|h_y\rangle|^2 \leq 1/\mathrm{poly}(n)$ for the NO case
⇒Some nodes must reject w.h.p.

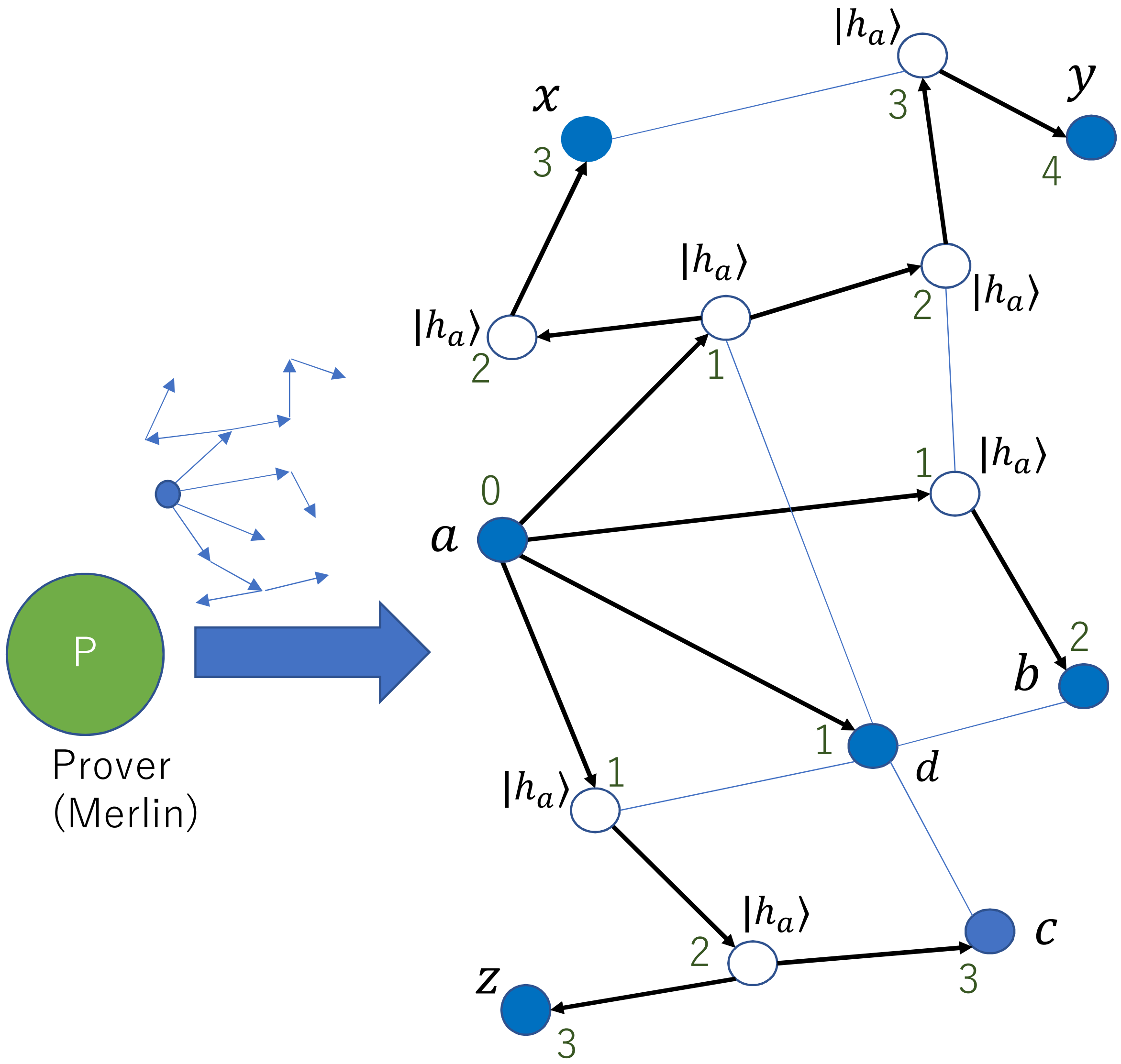
Verification phase

1. Each node j (except right node) chooses $b_j \in \{0,1\}$ uniformly at random: if $b_j = 0$, j sends the state to the right neighbor; otherwise, keep it by itself.
2. Each node (except left node) does SWAP test if it has two states, and outputs its result (accept/reject), and accepts otherwise



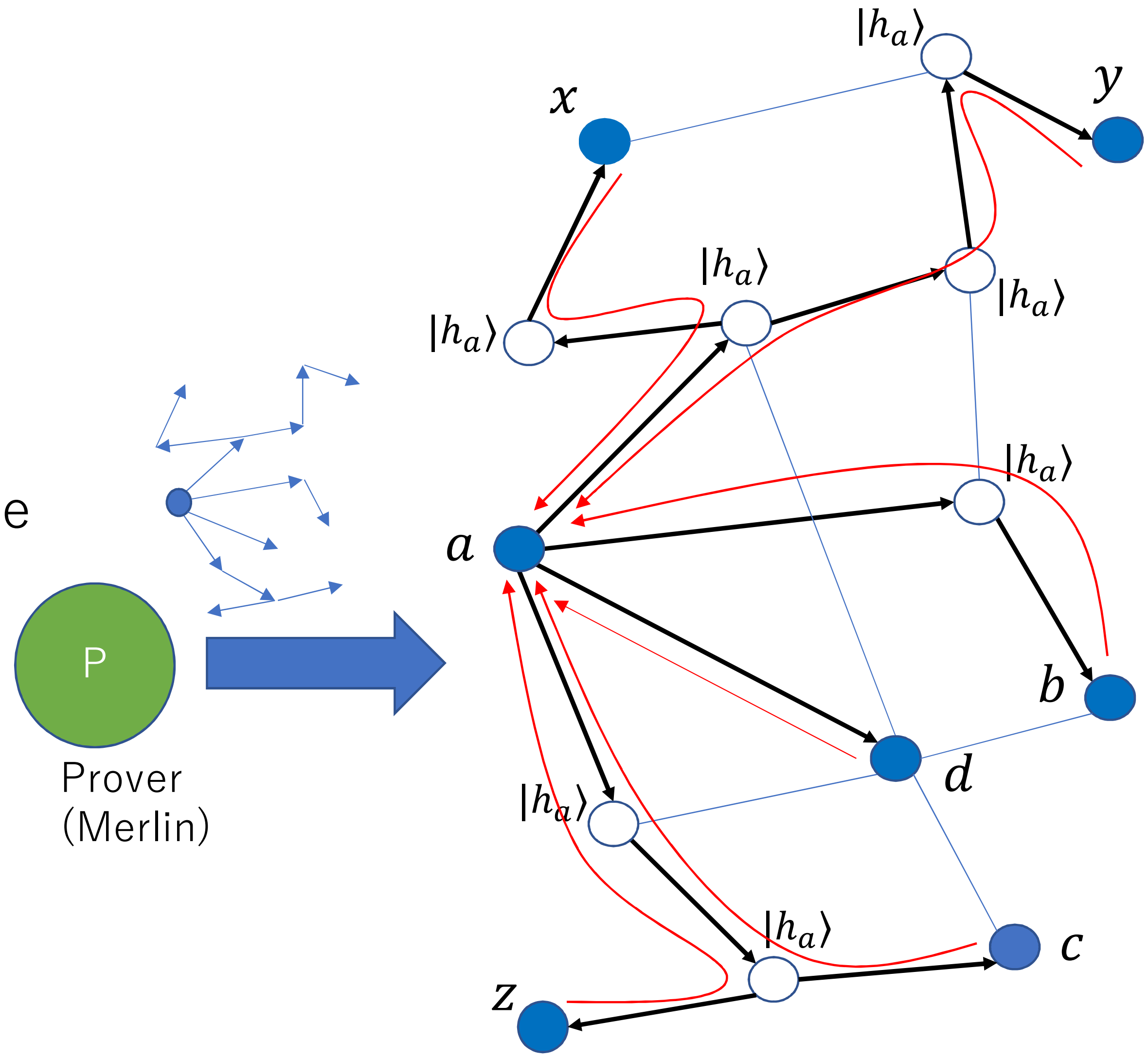
General Graphs

- Merlin sends a rooted tree with fingerprints:
  - Root is a terminal
  - Leaves are the other terminals



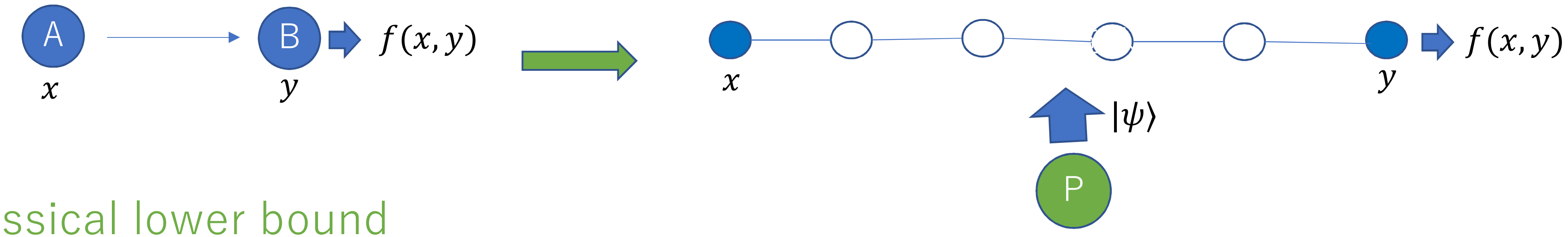
General Graphs

- Merlin sends a rooted tree with fingerprints:
  - Root is a terminal
  - Leaves are the other terminals
- Run the protocols on lines from the root to terminals in parallel



Our Results

- Distributed Quantum Merlin-Arthur (dQMA) protocols
  - Quantum certificates from the prover & quantum messages among nodes
- Quantum upper bound
  - ∃ dQMA protocol for equality of replicated data with $O(tr^2 \log(n+r))$-qubit certificates & messages (t:= number of the terminals; r ≔ radius of the network)
  - Extends to a **more general protocol** that converts one-way quantum communication complexity protocol to dQMA protocol **in line graphs**

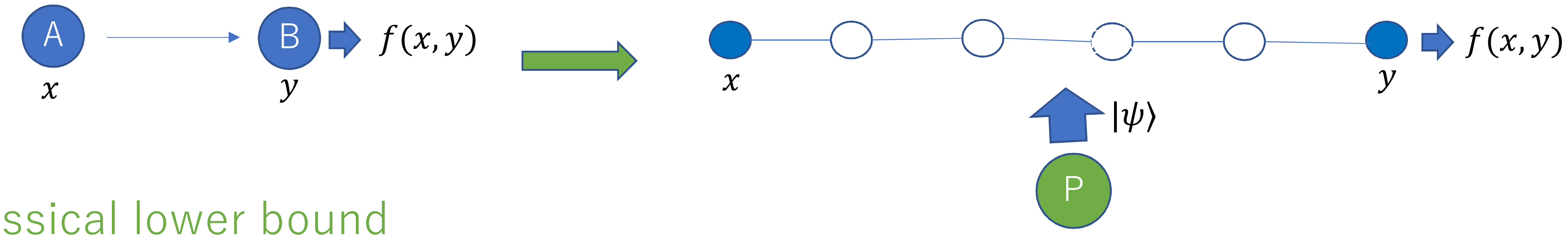


- Classical lower bound
  - Any dMA protocol requires $\Omega(n)$-bit certificates if error probability is reasonably small (say, 1/3)

Future work

- Distributed Quantum Merlin-Arthur (dQMA) protoc
  - Quantum certificates from the prover & quantum messages ar odes
- Quantum upper bound
  - ∃ dQMA protocol for equality of replicated data with $O(tr^2 \log(n+r))$-qubit certificates & messages (t:= number of the terminals; r ≔ radius of the network)
  - Extends to a **more general protocol** that converts one-way quantum communication complexity protocol to dQMA protocol **in line graphs**

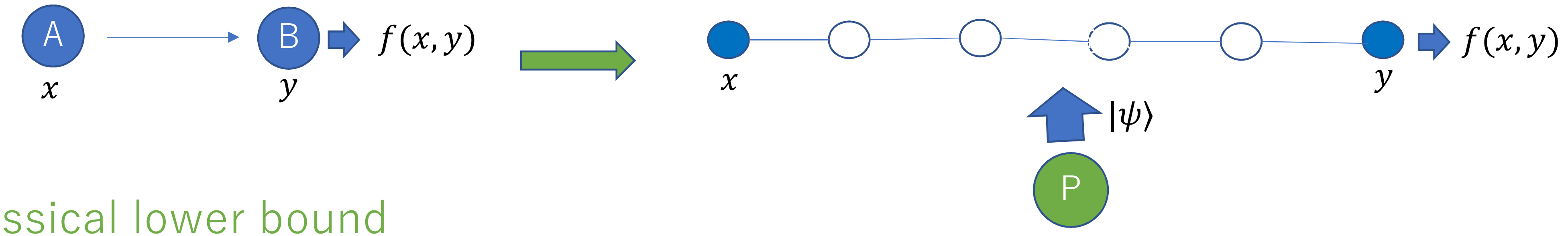
Q1: Is there a dQMA protocol better than dMA in terms of the graph size parameter?



- Classical lower bound
  - Any dMA protocol requires $\Omega(n)$-bit certificates if error probability is reasonably small (say, 1/3)

Future work

- Distributed Quantum Merlin-Arthur (dQMA) protoc
  - Quantum certificates from the prover & quantum messages ar odes
- Quantum upper bound
  - ∃ dQMA protocol for equality of replicated data with $O(tr^2 \log(n+r))$-qubit certificates & messages (t:= number of the terminals; r ≔ radius of the network)
  - Extends to a **more general protocol** that converts one-way quantum communication complexity protocol to dQMA protocol **in line graphs**



- Classical lower bound
  - Any dMA protocol requires $\Omega(n)$-bit certificates if error pr small (say, 1/3)

Q1: Is there a dQMA protocol better than dMA in terms of the graph size parameter?

Q2: Any quantum lower bound?